

Het pakket is gekozen. Jawel, dit pakket is zo flexibel, dat al onze wensen te realiseren zijn. Pakketspecialisten gaan samen met de gebruikers de nieuwe applicatie vormgeven. Moeten we nu nog functionele specificaties maken en zo ja, hoe?

Goed omgaan met pakketsoftware

Het opstellen van functionele specificaties

Nieuwe informatiesystemen worden in toenemende mate gebaseerd op pakketsoftware in plaats van maatwerksoftware. Leveranciers als SAP, Oracle, Siebel en Open Text bieden standaard oplossingen voor resource planning, content management enzovoorts, die in hoge mate configureerbaar en programmeerbaar zijn. IT-professionals, waaronder pakketspecialisten, zullen samen met gebruikers en andere belanghebbenden de juiste inrichting van een dergelijk pakket moeten bepalen. Om duidelijkheid te verschaffen over de vereiste functionaliteit, aan gebruikers, testers en pakketspecialisten, is het belangrijk om deze functionaliteit op z'n minst globaal te formuleren. Gangbare technieken voor functionele specificatie, die voor maatwerkprojecten zijn bedoeld, zijn in dit geval slechts in beperkte mate geschikt. Dit artikel gaat in op de specifieke problematiek van het specificeren

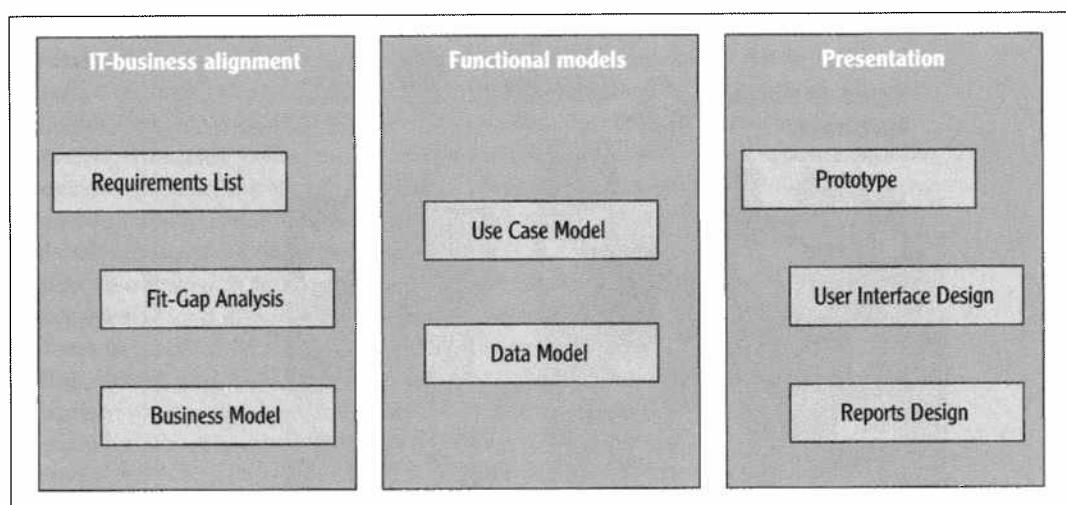
van pakketconfiguraties en pakketaanpassingen en biedt houvast bij het kiezen van een strategie.

In figuur 1 staat een aantal gangbare producten voor het vastleggen van specificaties afgebeeld. Deze komen in dit artikel aan de orde. Technieken voor informatievergaring (zoals workshops en interviews) worden niet behandeld, evenmin als de inbedding van het specificatiewerk in de ontwikkelmethode als geheel. Wel zullen we beginnen met een korte karakterschets van de aanpak. Daarna volgt een beschrijving van de specificatieproducten en tenslotte zullen we ingaan op de vraag hoe we door minder te specificeren meer rendement kunnen behalen.

Een aantal malen gebruiken we het woord 'pakketspecialist'. Meestal gaat het om een team van professionals, die het pakket goed kennen. Omwille



Hans Admiraal
is IT-architect bij Ordina.



Figuur 1: Specificatieproducten.

van de eenvoud is ervoor gekozen om specialisten met een functionele en specialisten met een technische focus onder één noemer samen te vatten.

De aanpak

In een maatwerksituatie hebben functioneel ontwerpers en gebruikers veel vrijheid om het systeem vorm te geven, ook al zullen er wel afwegingen gemaakt moeten worden tussen de functionele baten en de technische kosten. Bij een pakketconfiguratie daarentegen, moet men uitgaan van de functionaliteit van het pakket en de door het pakket geboden speelruimte benutten om er een volwaardige applicatie van te maken. Dit vergt veel pakketkennis én veel businesskennis. In teamverband zullen pakketspecialisten en materiedeskundigen (die de business vertegenwoordigen) elkaars kennis delen. De pakketspecialisten zullen met behulp van demo's en prototypen de gebruikers duidelijk maken wat het pakket kan en hoe zij adviseren de configuratie ervan vorm te geven. De materiedeskundigen zullen van hun kant feedback geven, maar ook proactief informatie aandragen, bijvoorbeeld over de gegevens die zij willen vastleggen en weer terug willen vinden op het scherm of in rapportages. Het resultaat van dit teamwerk wordt gedocumenteerd in de functionele specificaties. De dominante rol (in positieve zin) van pakketspecialisten en van prototypingtechnieken is kenmerkend voor het specificatieproces bij pakketsoftware.

Het vormen van een applicatie op basis van een pakket is meestal een evolutionair proces, waarin de producten (zie figuur 1) deels gelijktijdig worden gefabriceerd. Door de tijd heen convergeert men dan van het schetsen van de grote lijnen tot aan het invullen van de details.

IT-business alignment

Onder de noemer IT-business alignment vinden we in figuur 1 drie producten: Requirements List, Business Model en Fit-Gap Analysis.

Requirements List

In de meest eenvoudige vorm is de Requirements List een tabel waarin op elke rij een eis (of wens) staat en waarin de kolommen gebruikt worden om voor elke eis een aantal eigenschappen te definiëren: prioriteit, belanghebbenden en dergelijke. In grote projecten is het verstandig om hiervoor een requirements management tool te gebruiken, zoals IBM Rational RequisitePro of Teamcenter van Siemens. Op de Requirements List staan alleen eisen die door de belanghebbenden zelf zijn aangedragen, zonnodig hergeformuleerd om een goede kwaliteit te bereiken. De pakketspecialist zal aangeven welke eisen niet of slechts gedeeltelijk te verwezenlijken zijn met de standaard pakketfunctionaliteit. In die gevallen moet bepaald worden of de eis wordt geschrapt of afgezwakt, of dat de eis

door middel van een maatwerkuitbreiding wordt gerealiseerd.

Fit-Gap Analysis

De overeenkomsten en verschillen tussen de mogelijkheden van het pakket en de behoefte van de organisatie worden beschreven in de Fit-Gap Analysis. Dit is meer dan een opsomming van wat het pakket wel en niet kan. Bestaande processen en werkwijzen zullen moeten worden herzien om beter aan te sluiten bij het pakket. Op een creatieve wijze zal een visie moeten worden gevormd op de manier waarop van de ene kant het pakket geconfigureerd kan worden om het bedrijfsproces te ondersteunen en van de andere kant het bedrijfsproces heringericht kan worden om de implementatie van het pakket te bevorderen. Het is belangrijk dat de organisatie bereid is om zich aan te passen en bovendien zich neer te leggen bij bepaalde onmogelijkheden van het pakket. De Fit-Gap Analysis speelt dan ook de hoofdrol in de besluitvorming over het al dan niet inzetten van het pakket.

Business Model

Het Business Model beschrijft de bedrijfsprocessen die door het systeem ondersteund dienen te worden. Het definieert wie verantwoordelijk is voor welke taken en in welke volgorde deze taken worden uitgevoerd. In een verklarende woordenlijst wordt de business terminologie gedefinieerd. Meestal geeft een Business Model ook een globale beschrijving van de gegevens die in het bedrijfsproces een rol spelen. Een volledig Business Model beschrijft eerst de IST-situatie, die als input dient voor het maken van de Fit-Gap Analysis. Gelijktijdig met de Fit-Gap Analysis krijgt ook de SOLL-situatie in het Business Model vorm, dat is de toekomstige situatie, na invoering van het pakket. Gedurende het project zal dit model nader gedetailleerd en gecorrigeerd worden.

Functionele modellen

Met behulp van modellen (tekstueel en/of schematisch) kan een abstracte weergave van de functionaliteit van het systeem gegeven worden. We noemen hier alleen de populairste modellen: het Use Case Model en het Data Model.

Use Case Model

Een use case beschrijft hoe een gebruiker een bepaalde taak met behulp van het systeem kan uitvoeren. Deze taak is al geïdentificeerd in het Business Model, waarmee direct de context van de use case binnen het bedrijfsproces is gegeven. Als de use cases niet exact één-op-één gerelateerd zijn aan taken, dan zal in het Use Case Model uitgelegd moeten worden hoe de relaties tussen de taken en de use cases precies lopen.

Voor maatwerksoftware worden use cases vaak tot

Het vormen van een applicatie via een pakket is meestal een evolutionair proces.

Omdat de meeste functionaliteiten bij pakketten al standaard geleverd zijn, zijn ze erg gewild.

in de details gespecificeerd. Dit is voor pakketgebaseerde systemen niet raadzaam, omdat zulke details meestal de standaardfunctionaliteit van het pakket betreffen, terwijl we ons juist willen richten op de toepassings specifieke aanpassingen. Een korte beschrijving van het verloop van de mogelijke scenario's is voldoende (een stroomschema en gemiddeld één of twee pagina's tekst per use case). Functionele details zijn te vinden in het Data Model en het User Interface Design.

Data Model

Elk pakket heeft zijn eigen gegevensstructuur. Zo zal een CRM-pakket de entiteit 'Klant' kennen, met bijvoorbeeld de attributen 'Naam', 'Adres' en 'Telefoonnummer'. Het pakket biedt mogelijkheden om bepaalde attributen weg te laten, toe te voegen, een andere naam te geven, en de mogelijke waarden van een attribuut in te stellen. Deze gegevens spelen in meerdere use cases een rol, vandaar dat ze in een centraal Data Model gespecificeerd worden. Dit is een logisch gegevensmodel, hetgeen wil zeggen dat de gegevens worden gespecificeerd zoals de gebruiker ze ziet.

Presentatie

De meest concrete functionele specificaties zijn georiënteerd op de wijze waarop het pakket zich aan de gebruiker presenteert. In dit kader komen drie producten aan de orde: het Prototype, het User Interface Design en het Reports Design.

Prototype

De prominente rol van prototyping is in de paragraaf 'Aanpak' al genoemd. Een prototype is een beperkte inrichting van het pakket, die bedoeld is om een concreet beeld te geven aan gebruikers. Een beter begrip leidt tot betere feedback en dus een beter resultaat.

User Interface Design

In dit type ontwerp worden alleen de aanpassingen aan de user interface van het pakket beschreven. Hoe flexibeler het pakket, hoe meer men geneigd zal zijn de schermen aan te passen aan de wereld van de gebruikers. Welke velden en welke knoppen getoond worden, welke kolommen er in welke tabellen verschijnen, welke menuopties er zijn, het is vaak allemaal eenvoudig instelbaar. Het is niet efficiënt om dit alles van tevoren vast te leggen op papier, nee, de user interface wordt direct in het pakket vormgegeven. Gedurende het project vinden workshops met gebruikers plaats, waarin de schermen worden getoond en ongetwijfeld veelvuldig wijzigingen worden overeengekomen. Als de user interface (of een deel ervan) stabiel is, kan deze worden gedocumenteerd in het User Interface Design document, zodat alle afspraken duidelijk op papier staan en het systeem kan worden getest.

Reports Design

De meeste toepassingen vereisen ook rapportages. Deze komen analoog aan de User Interface tot stand. In het Reports Design staan alle benodigde rapportages gespecificeerd.

Andere producten

Afhankelijk van de toepassing kunnen er ook andere producten van belang zijn. Voorbeelden hiervan zijn: een Security Model, een Service Model en een Data Migration Specification.

Pakketten zijn gewild, omdat de meeste functionaliteit daarmee al standaard geleverd wordt. Het heeft uiteraard geen zin om al deze functies te specificeren. Wel moet vastgelegd worden op welke wijze het pakket moet worden ingericht naar de specifieke toepassing. Het volstaat echter niet om louter de verschillen tussen het pakket as is en de daarmee te realiseren applicatie te beschrijven. Ook standaardfuncties moeten geplaatst worden in de context van het bedrijfsproces, zodat duidelijk wordt op welke wijze het systeem de medewerkers ondersteunt bij hun werk, oftewel hoe medewerkers geacht worden met het systeem te werken. Dit is belangrijke input voor de testcyclus, voor het maken van opleidingsmateriaal en voor het veranderingmanagement.

Hoe zorgen we er nu voor, dat enerzijds onnodig specificeren van standaardfunctionaliteit wordt voorkomen en anderzijds wel een goed beeld ontstaat van het gebruik van het pakket in een specifieke bedrijfscontext? Dit is mogelijk, door een juiste balans te vinden langs drie dimensies: onvolledigheid, aantal producten en diepgang.

Onvolledigheid

Een belangrijk middel om niet teveel te specificeren is het bewust creëren van onvolledigheid: witte vlekken in de producten. Voor functies waarvan bekend is dat het pakket ze standaard al biedt, worden geen eisen in de Requirements List opgenomen en worden geen use cases geschreven. Het Data Model beschrijft alleen de toevoegingen en niet de reeds bestaande gegevensstructuren. In het User Interface Design blijven de ongewijzigde schermen achterwege. Al deze witte vlekken moeten het liefst wel benoemd en gemotiveerd worden.

Aantal producten

Voor elk van de functionele specificatieproducten zal ingeschat moeten worden of het nut ervan opweegt tegen de kosten. Talloze projecten zijn succesvol afgerond met functionele documentatie die alleen bestond uit een Use Case Model of alleen uit een User Interface Design. Als het pakket nog niet eerder in de organisatie is ingezet, dan is het verstandig om ervan uit te gaan dat alle genoemde producten nodig zijn en pas te gaan schrappen als is

aangetoond dat bepaalde producten overbodig zijn. Een voorbeeld. Een organisatie wilde een CRM-pakket inzetten om alle informatie ten aanzien van klanten en instanties waarmee dat bedrijfsonderdeel contact had, goed te administreren. Om welke gegevens het ging, wist men goed te benoemen, maar de bedrijfsprocessen die gebruik maakten van deze gegevens waren zeer divers en voor een belangrijk deel onduidelijk en ad hoc. We hebben er toen voor gekozen geen Business Model en geen Use Case Model te maken.

Diepgang

Tenslotte kan gestuurd worden langs de derde dimensie: diepgang, detailniveau. Van het Use Case Model is eerder al betoogd, dat dit het beste vrij globaal kan blijven. Ook een prototype hoeft niet in detail te kloppen. Voor het Data Model, het User Interface Design en het Reports Design is het gevaarlijker om te bezuinigen op diepgang. Om duidelijkheid te verschaffen aan met name ontwikkelaars en testers, is het belangrijk om precies vast te leggen hoe de aanpassingen aan het pakket eruit moeten zien.

Niet doen

Pakketspecialisten hebben vaak de neiging om één ontwerp te maken, waarin zowel de functionele als de technische aspecten van de vereiste pakketinrichting door elkaar staan. Dat is zeker efficiënt en duidelijk genoeg voor de specialist zelf, maar niet voor de materiedeskundige en de tester, tenzij deze zelf ook pakketspecialist is. Het is onwenselijk dat de specificaties slechts voor een elite toegankelijk zijn. Deze tactiek moet dan ook afgeraden worden.

Relevante factoren

Uit het voorbeeld blijkt al, dat de aard van het probleem en van de organisatie voor een deel bepalen

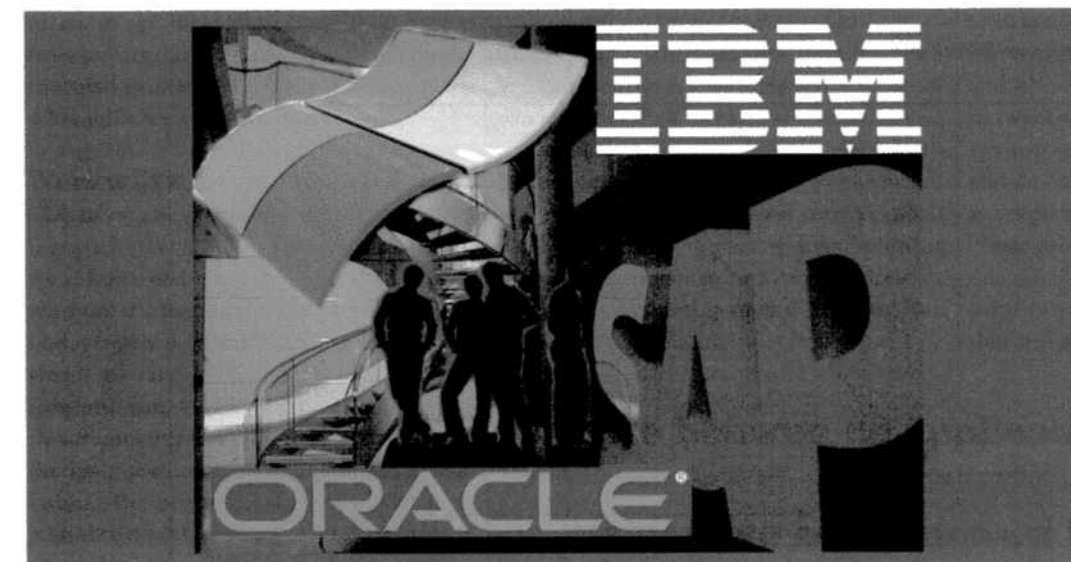
hoe de specificaties er uitzien. Maar er zijn meer factoren om rekening mee te houden:

- De teamgrootte. Hoe groter het team, hoe meer er gespecificeerd moet worden. In een klein en hecht team blijft iedereen mondeling op de hoogte en kunnen misverstanden snel gecorrigeerd worden.
- De complexiteit van het project. Een complex probleem vergt een grondige en systematische analyse. Bij eenvoudige toepassingen kan men met prototyping snel tot resultaten komen.
- Het vertrouwen tussen de betrokken partijen. Als er veel bedrijfsonderdelen betrokken zijn bij het project, die zich snel miskend of benadeeld voelen, dan is het verstandig om gedetailleerde specificaties op te stellen en te laten accorderen. Is iedereen daarentegen overtuigd van het gemeenschappelijk belang, dan scheelt dat papier.

Zo zijn er meer factoren, waarover al uitgebreid geschreven is in de vakliteratuur. Deze zaken zijn niet specifiek voor pakketsoftware, maar gelden voor maatwerk evenzeer. Vandaar dat ze in het kader van dit artikel niet verder worden uitgewerkt.

Conclusie

Functionele specificaties spelen een cruciale rol in elk IT-project. Het is lastig om te bepalen welke functionele producten er precies gemaakt moeten worden en met welke diepgang. Het is verstandig om een ervaren consultant op dit gebied te laten meedenken over de aanpak. Dit artikel kan een kapstok zijn om die discussie te voeren, zodat het project niet verzandt in eindeloze specificaties of eindeloze klachten van gebruikers, maar een passend eindproduct levert tegen redelijke kosten. «



IT-professionals moeten samen met gebruikers en andere belanghebbenden de juiste inrichting van het aangeschafte pakket bepalen.

Het is verstandig een ervaren consultant mee te laten denken bij het maken van functionele producten.